Journal of
Data Mining in Genomics & Proteomics

Zhou et al,. J Data Mining in Genom Proteomics 2012, 3:2
DOI: 10.4172/2153-0602.1000115

Research Article Open Access

OmicsMiner: A Biological Data Mining Framework

Zhenguo Zhou[1], Michael Netzer[2], In-Hee Lee[1], Michael Handler[2], Vijay Anand Manickam[1], Christian Baumgartner[2], Gerald H. Lushington[1] and Mahesh Visvanathan[1]*

[1]*Bioinformatics Core Facility, University of Kansas, Lawrence, KS 66047, USA*
[2]*Institute of Electrical, Electronic and Bioengineering, University for Health Sciences, Medical Informatics and Technology (UMIT), Tirol, Austria*

Abstract

OmicsMiner is a computational platform providing systematic access to state-of-the-art data processing and mining methods with the goal of facilitating the design of customized pipelines for processing a diverse range of biological data sets. Many built-in methods are provided for preprocessing, feature selection, clustering and classification of complex datasets. The platform supports convenient integration of additional algorithms that can further expand its functionality. OmicsMiner also provides convenient and concise interactive graphical user interfaces for data processing. OmicsMiner is a Java program that is platform-independent and does not require installation. It is available at http://www.bcf.ku.edu/software.shtml

Introduction

Since the completion and publication of the Haemophilus influenzae genome sequence in 1995 [1], systems biology has marched into a new phase featuring comprehensive analysis of biological systems. High-through put experimental studies such as genomics, proteomics, metabolomics, etc., (collectively referred to as omics) have transformed molecular cell biology from a field in which one gene or protein is studied at a time into a data-intensive pursuit in which whole organelles and pathways are studied simultaneously. A variety of omics sub-disciplines, each with its own set of instruments, techniques and software has begun to emerge. The omics technologies that have driven these new areas of research include DNA, RNA and protein microarrays, mass spectrometry, next generation sequencing and a number of other platforms that enable high-throughput molecular analyses [2]. Numerous algorithms and workflows have arisen to analyze omics data. The increasing complexity of the algorithms and the changeable workflow for analysis impose significant challenges for the analysis process. Fast deployments of new algorithms as well as assembly of predefined, easy-to-apply methods are important requirements for an omics data analysis framework. OmicsMiner is an organized collection of state-of-the-art data preprocessing and mining methods, and places an emphasis on simplifying the practical employment and combination of preprocessing methods and analysis algorithms.

Background

High-throughput analysis technologies such as DNA/RNA/protein microarrays, mass spectrometry and next generation sequencing provide a large amount of various types of biological data for bioinformatics research [2]. As a result, a variety of omics disciplines, each with its own set of instruments, techniques and software, has emerged and numerous algorithms and methods have been proposed to handle these types of data. Defining a globally applicable protocol for addressing omics data is challenging since the precise data format and processing requirements varies for each discipline, and the decisions about using which methods to apply (and in what combination) depend on the type of data, purpose of analysis and observations made during the analysis itself. Thus, instead of a single standard pipeline, there would be great value in having an integrative framework capable of handling a broad range of different data forms, supporting the flexible design of customized analysis workflows and facile deployment of new algorithms.

Motivation

The tremendous success of new high-throughput technologies in omics disciplines is mostly due to their unprecedented capability to acquire an ever growing number of analytical measurements (which analysts refer to as features) in rather low-cost experiments. This allows for the simultaneous investigation of a large number of genomic (or other molecular) loci. However, along with the huge number of features, another common characteristic of those biological data is a limited number of distinct samples.

Those two major characteristics of omics technologies present a major problem for mining the data: how does one find the features that are most sensitive and relevant to phenomena of interest? To achieve this, numerous methods have been developed for the purpose of filtering of the large data matrix and eliminating the superfluous features. The aim of these filters is to find a sub-matrix whose information content largely encapsulates the whole original matrix. Some filtering methods have been developed specifically for a single form of omics data, while others are more general. Some require corresponding specific preprocessing steps and some need to be combined with other particular methods for better performance. Most of these methods are available only as stand-alone programs or proof-of-concept implementations. Nevertheless, there is frequently a tangible benefit for combining several of these methods to process the results of a normal experiment. The choice of which methods to use and in what order depends on the nature of the data, the experimental conditions and on observations made during the analysis itself. Thus, bioinformaticians need an integrative framework which can process heterogeneous datasets, provide access to commonly (and easily) applicable algorithms, support convenient deployment of new algorithms, facilitate specification of specific pipelines for different data types, and enable cross-validation across different algorithms or pipelines.

***Corresponding author:** Mahesh Visvanathan, Bioinformatics Core Facility, University of Kansas, Lawrence, KS 66047, USA, E-mail: mvisvanathan@ku.edu

Received November 19, 2011; **Accepted** March 23, 2012; **Published** March 25, 2012

Citation: Zhou Z, Netzer M, Lee IH, Handler M, Manickam VA, et al. (2012) OmicsMiner: A Biological Data Mining Framework. J Data Mining in Genom Proteomics 3:115. doi:10.4172/2153-0602.1000115

Related work

There has been research effort directed towards this end and various applications have been developed for addressing a range of data types [3-8]. However most of those approaches adhere to a narrow focus dictated by discipline-specific requirements. There do exist several general analysis frameworks such as BioWeka [9], Galaxy [10], Bioinformatics Computational Journal [11] and Taverna [12] or libraries like Bioconductor [13], Bioperl [14] and KD3 [15] that can have a more diverse range of data forms. Several of these frameworks are briefly described below. MultiExperiment Viewer (MeV) is a microarray data analysis tool inside the TM4 suite. It is an open-source software including algorithms for clustering, visualization, classification, statistical analysis and biological theme discovery [8]. BioWeka [9] is an extension library implemented within the Weka framework [16] that supports biological data formats and alignments and is licensed under GNU General Public License. EMMA2 [4] is an open-source platform to store and analyze microarray data. Its database can manage raw experimental data and relevant biological and technical information. It is a collection of various algorithms including normalization and pre-processing, statistical inference, cluster analysis, data integration and visualization features. Automatic analysis is also enabled by specification of pipelines. Gene Pattern [6] is a genomic analysis web application for gene expression analysis, proteomics, SNP analysis and some other common data processing tasks. It also allows the creation of multi-step pipelines. KD3 (Knowledge Discovery in Databases Designer) is a workflow-oriented software suite covering the complete database knowledge discovery process and offers a variety of implemented methods, algorithms and workflows that can easily be extended by adding new customized components (so called functional objects) [15].

Contribution

Customization of these frameworks or libraries for user-specific applications requires the development of interface utilities or web-based applications which handle data outside of a local system. This requirement can be a burden for biologists, and web-based analysis can raise data security issues. Therefore we propose a general data analysis framework, called OmicsMiner, which can handle a variety of biological data forms within local system using a flexible workflow design to enable customization without the need for complex programming. OmicsMiner implements an organized collection of state-of-the-art data processing and data mining algorithms that address preprocessing, feature selection, clustering, classification and visualization (Figure 1). Additional algorithms can easily be included into the framework to expand the functionality of OmicsMiner provided that they are written in compliance with its interfaces. The OmicsMiner architecture has been designed with an emphasis on flexibility; supporting a broad range of biological data formats and enabling specification of customized workflows that combine various existing and new analysis algorithms. OmicsMiner is distributed as a stand-alone Java application with a graphical user interface for easy manipulation and data management. Since KD3 [15] and Weka [16] classes provide basic common functionalities, many components of OmicsMiner are based on KD3 functional objects and Weka classes. However we have augmented the suite with new utilities suited for handling different types of omics data not supported in KD3 and Weka.

The main features of omicsminer can be summarized as follows:

Flexibility in pipeline design: Users can freely design specific workflows for a data set of interest. Since each algorithm in the framework is modular, any new workflow can be defined by choosing and ordering available modules. The designed workflow can be saved for later use or further adaptation.

Capability to handle broad range of data type: OmicsMiner can handle any type of numerical data set as input if it is provided in one of the supported formats described in Section 2.3. It also supports the raw microarray image format (.CEL format) and an abstract data format shared among data mining tools. It also supports convenient export of data and processed information in a variety of different formats.

Modular implementation with Java beans: All operators are modularized and implemented as Java beans and can be easily configured during runtime. New algorithms can be incorporated into the framework with little effort if they follow the relevant interfaces.

Data visualization: OmicsMiner provides a variety of common ways to visualize data so that user can easily analyze and interpret the data.

Outline

This remainder of this work is organized as follows. Section 2 describes the OmicsMiner system in detail, beginning with a discussion of the central system features and an overview of the pipeline architecture. Pipeline components are discussed in detail and typical usage patterns are summarized. Section 3 presents a series of experimental studies performed using OmicsMiner. Each study is described in terms of the data sources, feature sets, and the experimental protocols observed. Results from studies are evaluated in terms of predictive performance and selected feature relevance.

Methods

This section describes the OmicsMiner system in detail. First the salient features of OmicsMiner are presented and then an overview of the system architecture is given. The major aspects of OmicsMiner are then reviewed: data formats, preprocessing, feature selection, and data mining (including classification, clustering, statistical analysis, and visualization). Finally, common activities in OmicsMiner are briefly described.

System features

The OmicsMiner system boasts several key features to manage data and streamline analysis, described here.

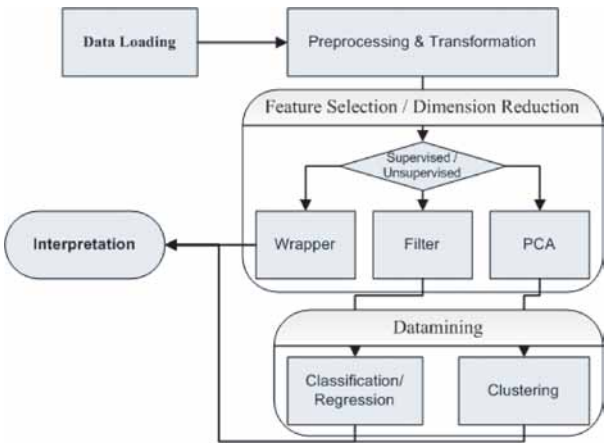

Figure 1:

Data-oriented: Using a data-oriented design, the focus of programming is on the data itself. The ordering of operations is not explicitly specified in the framework, but is defined by the analyst who uses it and by the requirements of the data. Functionality in this framework entails transferring data from one functor object to another object and from one kind of format to another format. These functor objects are ideally suited for processing the input data and creating some specific kind of output data.

Functor objects: All the functionalities and algorithms in this framework are implemented as objects. One object is designed for each unique algorithm. These objects can be easily added into a pipeline and assembled together into the framework and implemented along with current objects. In OmicsMiner, all the functor objects are implemented as Java beans and can be easily configured during runtime of the application. Thus, new algorithms can be readily combined into the framework provided that adhere to the Java beans criteria and are configured according to the relevant interfaces of the framework.

Pipeline design for specific dataset: While microarray data is not the only form that OmicsMiner can handle, it is a common medium for high throughput experiments, thus our design has been tailored to support a variety of different microarray formats. For microarray data, there are different supported analytical platforms including DNA, oligonucleotide, SNP, MMChips, Protein, Tissue, Cellular, Chemical compound, Antibody, and Carbohydrate arrays (glycoarrays). Assembly of objects into a customized sequence (i.e. a pipeline) permits one to preprocess, filter, mine, validate and visualize each data format in a manner suitable for that format. Pipeline design and reuse is thus a major feature of the whole OmicsMiner framework architecture; users who design and optimize specific pipelines for particular datasets can save those predefined pipelines as templates for later analysis or other applications.

Different views of data: In OmicsMiner, users can view data in several different forms, including data table, feature summary, graph view, execution results. Additional data view panels can later be easily combined into this framework.

Pipeline overview

OmicsMiner supports the design and configuration of pipelines for automated streamed data processing. Additional algorithms can easily be combined into the framework to extend OmicsMiner's functionality provided that they are written using the interfaces implemented in the framework. This results in the availability of convenient and concise project-specific interactive graphical user interfaces for data processing. The most common protocols to analyze omics data generally include the following steps: preprocessing, inference or classification and validation [17].

Data formats

OmicsMiner allows the user to import data from different types of data file formats since biological data comes in many distinct formats. So far, the following data formats are supported:

ARFF: ARFF is a special format Weka uses for its datasets. Since it is commonly used in several major data mining tools, we support it in OmicsMiner.

CSV and excel: Large amounts of biological data are stored in these kinds of formats. The framework can also load files containing tab-separated values.

CEL: Cell Intensity File. The .CEL files contain the information from the scanning images of the hybridized microarrays. OmicsMiner allows user to read multiple CEL files as a microarray data matrix and to add the class labels to each sample for using the following facilities to analyze the experiment data.

In order to be imported correctly, the input data should correctly adhere to one of the above format specifications. Using a well-formed data file as input will result in correct implementation of the pipeline and will yield valid output. Some limitations may exist on the size of the data set that can be imported; however. The maximum number of samples that can be loaded into OmicsMiner at one time will depend on the available RAM assigned to JVM in the computer running OmicsMiner and the number of features available to characterize each sample.

Generally, there are two kinds of data which can be analyzed in OmicsMiner. One kind has class label information, which can be used in classification and supervised feature selection. The other lacks class label information, and can be used in clustering and unsupervised feature selection. Normally, for csv and excel files, OmicsMiner needs data to be in the form of a matrix with feature labels. Sample identifiers may be present but will not be used in the program.

Preprocessing

OmicsMiner supports common dataset preprocessing methods such as feature or sample normalization, Log-transformation and Z-Transformation. Some basic data set handling methods are also included in the framework, such as the ability to sort by row or column, and merge data sets from different data files. For the merge component, if the mergeCols property is true, the datasets must have the same number of features; if mergeCols is false, the datasets must have the same number of samples. For raw microarray intensity data, OmicsMiner implements standard preprocessing steps integrated in loading process: background correction, normalization, pm correction, and summarization. For each microarray preprocessing step, OmicsMiner provides multiple algorithms and the user can choose any combination of preprocessing methods when loading microarray files.

Feature analysis

Analysis of sample features, including selection and transformation, is a critical component of data mining. OmicsMiner includes a broad range of built in tools for this purpose. Here the available algorithms are presented and briefly described.

Feature selection: Selection of relevant features for data mining is a common task before proceeding with analysis of biological data. This is especially true in the case of gene expression levels, where the large number of features may degrade the performance of many analysis tools. The following feature selection approaches are available.

Biomarker identifier (BMI): the BMI, which was originally applied on metabolic data, combines various statistical measures such as the discriminatory performance, discriminatory space, and variance of metabolites' concentrations at the state of disease to calculate an evaluation score for feature ranking [18].

ANOVA (one way analysis of variance for every attribute): is a technique used to compare means of two or more groups (using the F distribution). This technique can be used only for numerical data [19].

Information gain: evaluates the worth of an attribute by measuring the information gain with respect to the class [20].

Kruskal-wallis feature selection: a non-parametric method to compare groups [21].

Relieff: Evaluates the worth of an attribute by repeatedly sampling instances and considering the value of the given attribute for the nearest instances of the same and different classes [22].\

Chisquared test: evaluating attributes individually by calculating the chi-squared statistic.

Feature selection evaluation: In Feature Selection Evaluation, we simultaneously calculate the scores for features using multiple feature selection algorithms like BMI, Information gain, reliefF to enable convenient comparison of the features scores.

Fold-change calculation is also combined in several feature selection methods for features scores' comparison.

Principal component analysis: When a reduction in the data dimensionality is important, but selection of individual features is not desired, data transformation is often employed. Principal Component Analysis (PCA) is a common tool for this purpose. PCA transforms a set of potentially correlated features into a set of orthogonal, uncorrelated ones. The transformed features are arranged by decreasing variance, thus selection of only the first few principal component features yields a data set that describes the original samples with much lower dimensionality.

Correlation analysis: OmicsMiner implements Linear Correlation Finder to analyze correlation between different features in the biological data set using Spearman and Pearson. OmicsMiner also used Correlation Matrix to calculate Spearman and Pearson correlation matrices among all the features. In both of cases, users need only set the α significance level, but can also set the attribute name filter to reveal the correlation coefficients for those specific features.

Data mining

While all pipeline components presented thus far are important, data mining is the ultimate goal of the OmicsMiner system. Here a number of mining and discovery tools included with OmicsMiner are described.

Classification and regression: Data classification or prediction refers to the process of learning a function that maps data samples to two or more discrete classes. Classification is studied by a wide variety of researchers for just as many purposes, and hence there exist numerous classification methods. OmicsMiner has included some of the most popular and relevant classifiers for omics data prediction. Regression is similar to classification except that the learned function maps a body of samples to a numerical trend instead of a discrete class.

Clustering: Clustering is one of the most common tasks in microarray analysis. OmicsMiner offers several clustering methods with different optimization criteria including the well-established partitioning methods such as k-means, hierarchical clustering and Ordering Points To Identify the Clustering Structure (OPTICS) [23] using the corresponding KD3 functional object. All clustering methods can be performed with a wide range of parameters such as choosing Manhattan distance or Euclidean Distance, setting the number of clusters in k-means, using single or complete linkage in hierarchical clustering, and so on.

Statistical testing: Outlier detection and normal distribution test are implemented in this part. Statistical tests include Student's t-test and ANOVA as well as non-parametric tests such as Wilcoxon rank sum and Kruskal-Wallis test.

Visualization

OmicsMiner supports some mode of visualization for the analysis associated with every functor object. This includes mapping dendrograms for hierarchical clustering, reporting feature correlation in Cartesian form, plotting a score histogram for each feature selection algorithm and cluster feature distribution graphs in clustering methods, etc. We also provide several specialized plotting functions for PCA analysis and ROC curves.

Using omicsminer

OmicsMiner is a general tool for data analysis; however the most common usage patterns are envisioned to include several fundamental activities. These basic OmicsMiner activities are described here in brief.

Starting omicsminer: Since OmicsMiner is developed in Java platform and distributed as a jar file, it can be started just by going to the directive and using command line "java -jar OmicsMiner". If user needs to process a large dataset, then -Xmx option should be used to specify the heap size for JVM and the actual size of that should depend on the configuration of user's computer.

OmicsMiner also combines some functionality in R, like loading CEL (for raw microarray images) formats files. In order to use this functionality, users need to add environment variables for the installing folders of R. To simplify OmicsMiner usage, a bash file has been written to serve this function. Having referenced the bash file, users can go to the OmicsMiner directory and use the command line /run.

Pipeline designing: Pipeline designing in OmicsMiner is very straightforward. After starting OmicsMiner, one selects the first functor object in left tree structure panel (normally it is LoadData), sets the parameters of this object and then clicks add to add it into the pipeline indicator panel. Figure 2 shows a sample pipeline consisting of three functor objects. The user can then select other functor objects and finish the configuration of a pipeline.

Pipeline modification: Modifying pipeline allows users to add or eliminate functor objects in current pipeline as well as to modify the parameters of a specific functor object.

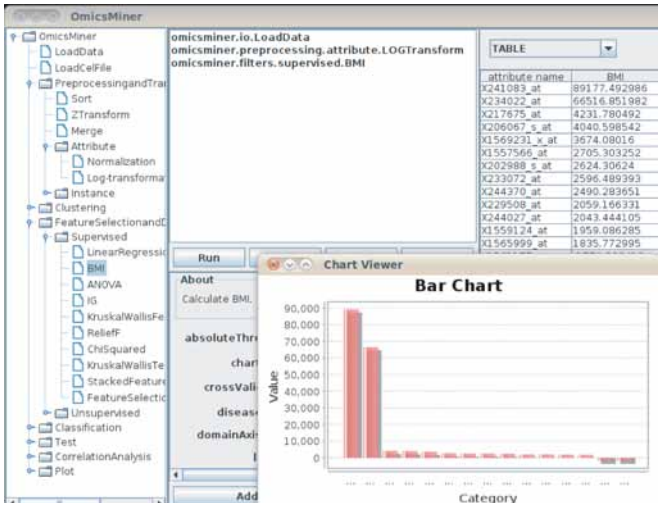

Figure 2:

Loading and saving pipelines: After creating or modifying a pipeline, users can just click *save* button to save the pipeline for future use. Later, the pipeline can be loaded using the load button.

Running the pipeline: When a pipeline is established, users can click run to execute the pipeline in a flow manner.

Saving the data: Users can save the result data set as a csv or xls file for future use.

View data from different aspect: User can have different views of data using the combo box on the top of data view panel.

Experimental Results

In order to evaluate the effectiveness of OmicsMiner for common data analysis tasks, we have performed a series of experimental studies highlighting several important components of the OmicsMiner architecture. Each study is described below, with discussion of data sources, features, experimental protocol and results.

Cancer cell feature selection

To demonstrate the flexibility of pipeline design in OmicsMiner, we first performed classification experiments on two cancer cell prediction data sets. Lung cancer accounts for the most cancer-related deaths and it is essential for effective treatment to identify lung-cancer-associated genes [24]. Two expression data sets were obtained and an OmicsMiner pipeline was used to perform feature selection on cell samples with the goal of distinguishing their phenotype (normal or cancerous).

Data sources: The first data set used contains gene expression levels obtained from GlaxoSmithKline (GSK), which has released the genomic profiling data for over 300 cancer cell lines via the National Cancer Institute's cancer Bioinformatics Grid® (caBIG®). The raw microarray data is available at https://cabig.nci.nih.gov/caArray_GSKdata/. We used data from 65 samples of small-cell carcinoma tissue and 41 samples of adenocarcinoma. The second data set is the Notterman Adenoma Dataset from Princeton University [25]. This dataset contains expression levels of 2000 genes taken from 62 different samples.

Feature selection: For both data sets, features represent gene expression levels. For the GSK data set, we compared BMI and ReliefF for feature selection after log-transformation. For the Notterman data, information gain, BMI, ANOVA, and KruskalWallis feature selection methods were employed.

Results: The feature selection results for the GSK data are presented in Figures 3 and 4. The ranking of features with BMI for this data produced two relatively high quality features, with the remaining features contributing less predictive ability. Using ReliefF to select features produced a distinctly different profile, with a large number of features ranked with relatively similar importance.

For the Notterman data set, Table 1 shows the top selected genes by ANOVA, BMI, Information Gain and KruskalWallis Feature Selection, and Figures 5 and 6 show the top ranked features by BMI and information gain. The results for this data set using BMI are again distinct from those obtained for the GSK data using BMI. Where the GSK data contained just a pair of highly ranked features, for the Notterman data BMI produced a number of features with similar if not identical scores. Information Gain, on the other hand produced a single high quality feature, while the remaining features were scored much lower.

Clustering gene expression levels

Clustering is another common task in omics data analysis. In this experimental study, we use the cellular gene expression levels to group cells according to phenotype.

Data sources: Gene expression levels were obtained from a public archive of the functional genomics database, ArrayExpressDB (http://www.ebi.ac.uk/arrayexpress/). The data used Affymetrix Rat Genome 230 2.0 platform. The investigated data set applied in this work comprises data of 7 individuals divided into two classes: control (n=3) and FSH-treatment (n=4). The number of measured gene expressions is 31,099.

Feature selection: BMI feature selection was used to reduce the very large dimensionality of this data set.

Clustering: Hierarchical clustering was performed on the gene expression levels of all individuals.

Results: From the Figure 7, we can see that hierarchical clustering

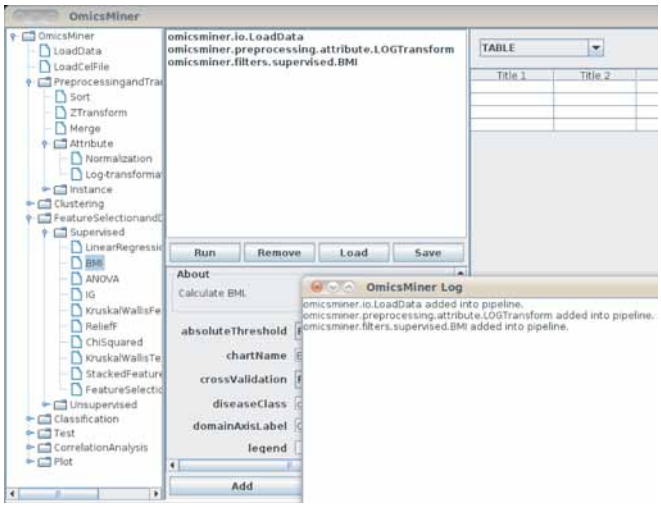

Figure 3:

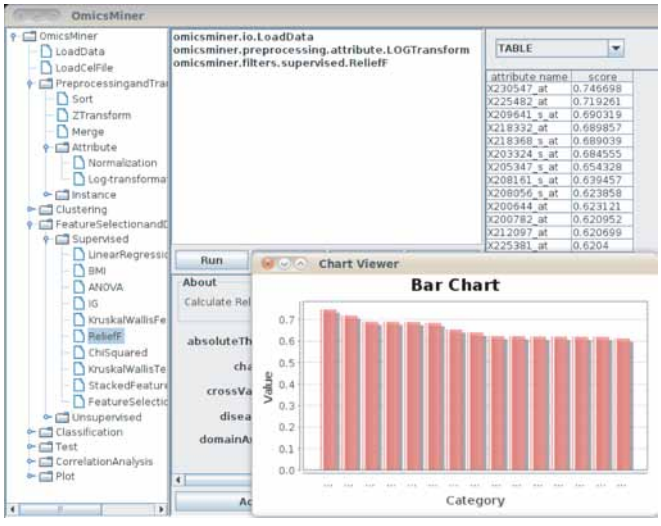

Figure 4:

ANOVA	BMI	IG	KWT
AF001548	U10099	AB000584	L08246
Z23090	X07820	AB000895	U02388
M12125	L40402	AB002365	U71374
X13839	X05839	AB002533	X16665
X15882	J03626	AB003698	AB000584
M11313	L20321	AF000959	AB000895
AB000584	X98225	AF001548	AB002365
M81757	M91368	AF005887	AB002533
M24194	U66464	AF006087	AB003698
L17131	L02785	AJ000480	AF000959
L25286	U57452	D00632	AF001548
X15880	HG2417-HT2513	D00654	AF005887
X00351	U24186	D10522	AF006087
J02783	L42379	D10667	AJ000480
D14662	X63454	D11428	D00632
Y00339	HG2981-HT3127	D13370	D00654
HG3044-HT3742	L20348	D13413	D10522
HG1612-HT1612	X92689	D13641	D10667
M61906	M26692	D13666	D11428
X14813	M57609	D13748	D13370
X78565	M58525	D14530	D13413
D87433	X99133	D14659	D13641
J04456	U04636	D14662	D13666
X01677	X85372	D14874	D13748
HG1153-HT1153	H46990	D15049	D14530
S82362	U41518	D16294	D14659
M34458	S67070	D17408	D14662
X74295	M87313	D21267	D14874
U50360	X61079	D23660	D15049
U12465	M83216	D23673	D16294

Table 1: Top Selected Genes for Notterman Data Set.

on selected features can differentiate the two classes with very high precision.

Feature selection for drug bioavailability

Bioavailability is a measure of the fraction of an administered pharmacological compound that reaches systemic circulation. Bioavailability depends on a range of different physiological and physicochemical effects, including compound solubility, capacity for intestinal absorption and amenability to distribution into plasma, thus modeling molecular bioavailability profiles must account for a variety of different attributes that affect molecular disposition within a variety of different biochemical environments. We arbitrarily classified compounds as being bioavailable if more than 50% of the dose was available. In this experimental study, we used OmicsMiner to select features relevant to the bioavailability of a number of chemical compounds.

Data sources

In our work, we used a data set reporting the human oral availability of nearly 800 compounds, as compiled by Hou et al. [26]. In our training set, we have total 580 chemical compounds with quantitative bioavailability data, and described each compound via 248 molecular features.

Feature selection

The feature set used to predict bioavailability profiles included a variety of structure-based and physicochemical properties computed from the Volsurf suite of volumetric and surface-projection descriptors [27], and the BCUT molecular diversity parameters [28]. We used OmicsMiner to calculate the BMI and ANOVA scores.

Results: Figures 7, 8 and 9 present the feature selection results for the bioavailability study. The results for feature selection with BMI indicate a number of features with similar importance. Analysis with ANOVA, on the other hand, yields a pair of high scoring features with the remaining features scoring much lower overall.

Discussion

The current version of OmicsMiner offers a combination of algorithms that can be applied at different stages of the data analysis process, with the aim of exploiting the simplicity of program interface and the comprehensiveness of the functor objects. The core structures have been optimized to improve performance and simplify the addition of new functionality, thus enabling analysis of different types of omics-data. Among the features of this framework are flexible data import and export options, the ability to create and save specific pipelines for given types of data, a simple and convenient user-interface, matrix operations

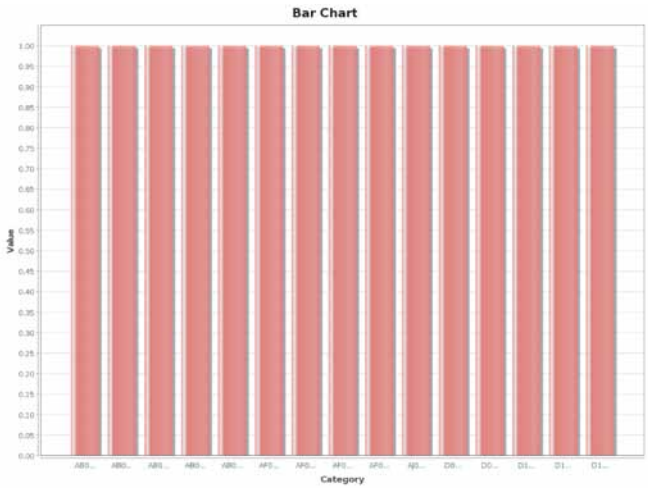

Figure 5:

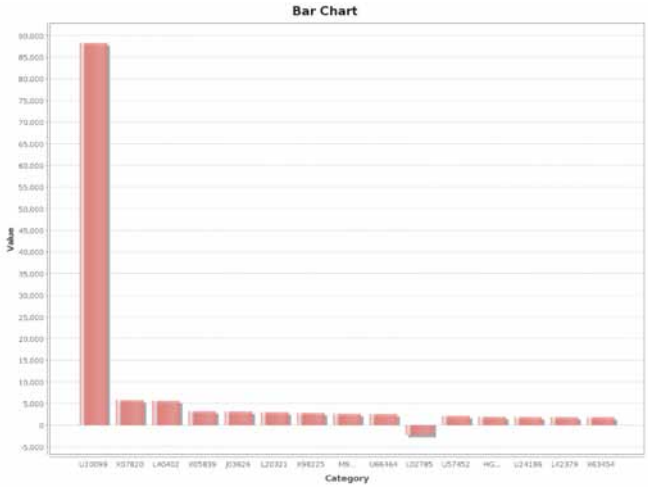

Figure 6:

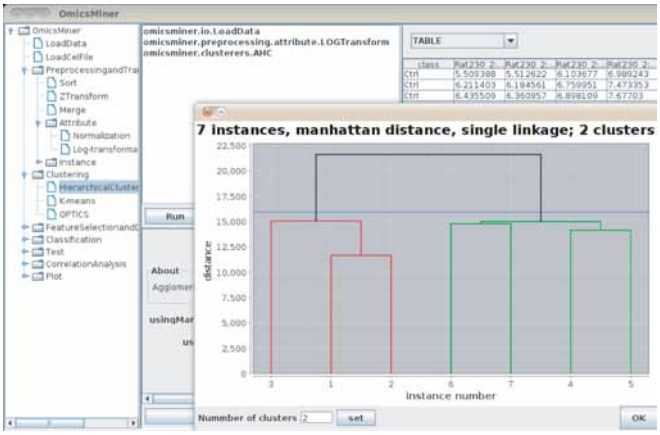

Figure 7:

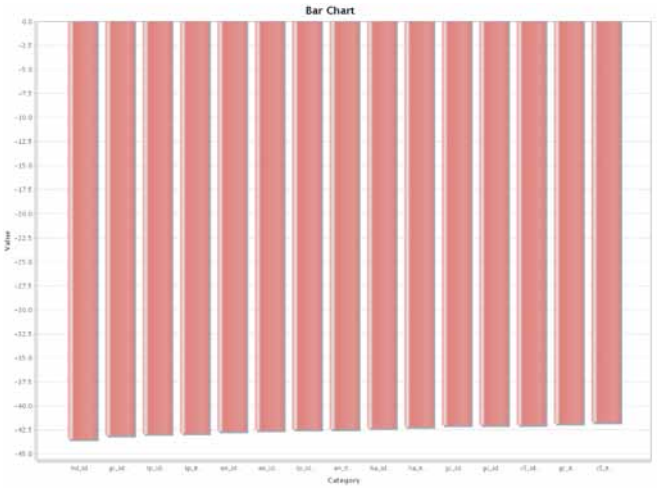

Figure 8:

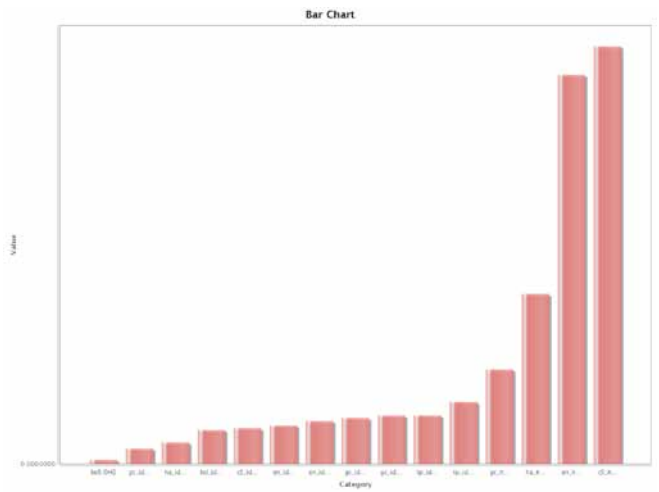

Figure 9:

for convenient set sorting and merging, new statistical methods for the identification of differentially expressed genes (or other applications with comparable data formats), online data transformations (e.g. ZTransforming, normalization and log-transformation) and many more.

Conclusion

OmicsMiner is a flexible framework for the analysis of various kinds of omics data. Biological data analysis methods ranging from preprocessing and normalization to novel statistical and machine learning methods have become highly sophisticated and new methods are published almost daily. The OmicsMiner framework can integrate such new methods and readily combine them into analysis pipelines, thus providing a convenient environment for biological data analysis. We have written OmicsMiner to instill flexibility for a diverse range of current applications, and for future expansion based on emerging algorithms and protocols. Among the large number of components are an automated processing framework, dynamic pipelines, and efficient feature selection methods. The suite consistently adheres to the aim of combining the simplicity of the program interface and the comprehensiveness of the functor objects toward effective and facile usage. Our future development will focus on direct support for next generation sequencing (NGS) data analysis. Currently, OmicsMiner can handle NGS data that has been pre-aligned and processed into a numerical matrix in csv or excel format. In future, we plan to enable OmicsMiner to load NGS data directly and interface with other NGS software packages such as BFAST [29], Bowtie [30], Cufflinks [31] and Genome Analysis Toolkit (GATK) [32] so as to make the pipeline more efficient and convenient.

Acknowledgements

This work was supported by award number P20 RR016475 from the National Center for Research Resources, by award number R01 HD061580 from the National Institute of Child Health & Human Development, and by the Austrian Genome Research Program GEN-AU (Bioinformatics Integration Network, BIN III). Also authors would like to thank Dr. Leslie Heckert for testing the program and giving valuable suggestions. We would also like to thank Dr. Aaron Smalter Hall in helping us in preparing this manuscript.

Author Disclosure Statement

No competing financial interest exists.

References

1. Fleischmann RD, Adams MD, White O, Clayton RA, Kirkness EF, et al. (1995) Whole-genome random sequencing and assembly of Haemophilus influenzae Rd. Science 269: 496-512.
2. Spiegelman D, Whissell G, Greer CW (2005) A survey of the methods for the characterization of microbial consortia and communities. Can J Microbiol 51: 355–386.
3. Biswas A, Mynampati KC, Umashankar S, Reuben S, Parab G, et al. (2010) MetDAT: a modular and workflow-based free online pipeline for mass spectrometry data processing, analysis and interpretation. Bioinformatics 26: 2639-2640.
4. Dondrup M, Albaum SP, Griebel T, Henckel K, Jünemann S, et al. (2009) EMMA 2 – A MAGE-compliant system for the collaborative analysis and integration of microarray data. BMC Bioinformatics 10: 50.
5. Qi J, Zhao F, Buboltz A, Schuster SC (2010) inGAP: an integrated next-generation genome analysis pipeline. Bioinformatics 26: 127-129.
6. Reich M, Liefeld T, Gould J, Lerner J, Tamayo P, et al. (2006) GenePattern 2.0. Nat Genet 38: 500-501.
7. Rieber N, Knapp B, Eils R, Kaderali L (2009) RNAither, an automated pipeline for the statistical analysis of high-throughput RNAi screens. Bioinformatics 25: 678-679.

8. Saeed AI, Sharov V, White J, Li J, Liang W, et al. (2003) TM4: a free, open-source system for microarray data management and analysis. Biotechniques 34: 374-378.

9. Gewehr JE, Szugat M, Zimmer R (2007) BioWeka – extending the Weka framework for bioinformatics. Bioinformatics 23: 651-653.

10. Goecks J, Nekrutenko A, Taylor J, Galaxy Team (2010) Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. Genome Biol 11: R86.

11. Feagan L, Rohrer J, Garrett A, Amthauer H, Komp E, et al. (2007) Bioinformatics process management: information flow via a computational journal. Source Code Biol Med 2: 9.

12. Oinn T, Addis M, Ferris J, Marvin D, Senger M, et al. (2004) Taverna: a tool for the composition and enactment of bioinformatics workflows. Bioinformatics 20: 3045-3054.

13. Gentleman RC, Carey VJ, Bates DM, Bolstad B, Dettling M, et al. (2004) Bioconductor: open software development for computational biology and bioinformatics. Genome Biol 5: R80.

14. Stajich JE, Block D, Boulez K, Brenner SE, Chervitz SA, et al. (2002) The Bioperl toolkit: Perl modules for the life sciences. Genome Res 12: 1611-1618.

15. Dander A, Handler M, Netzer M, Pfeifer B, Seger M, et al. (2011) KD^3: a workflow-based application for exploration of biomedical data sets. Transactions on Large-Scale Data- and Knowledge-Centered Systems 4: 148-157.

16. Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, et al. (2009) The WEKA data mining software: an update. SIGKDD Explorations 11: 10-18.

17. Allison DB, Cui X, Page GP, Sabripour M (2006) Microarray data analysis: from disarray to consolidation and consensus. Nat Rev Genet 7: 55-65.

18. Baumgartner C, Lewis GD, Netzer M, Pfeifer B, Gerszten RE (2010) A new data mining approach for profiling and categorizing kinetic patterns of metabolic biomarkers after myocardial injury. Bioinformatics 26: 1745-1751.

19. Howell DC (2002) Statistical Methods for Psychology, 7th ed. Wadsworth Publishing.

20. Kodaz H, Ozsen S, Arslan A, Gunes S (2009) Medical application of information gain based artificial immune recognition system (AIRS): diagnosis of thyroid disease. Expert Systems with Applications 36: 3086-3092.

21. Lan L, Vucetic S (2009) A multi-task feature selection filter for microarray classification. Proceedings of 2009 IEEE International Conference on Bioinformatics and Biomedicine (BIBM '09).

22. Kononenko I (1994) Estimating attributes: analysis and extensions of RELIEF. Lecture Notes in Computer Science 784: 171-182.

23. Ankerst M, Breunig MM, Kriegel H-P, Sander J (1999) OPTICS: ordering points to identify the clustering structure. Proceedings of the 1999 ACM SIGMOD international conference on Management of data.

24. Jemal A, Siegel R, Ward E, Hao Y, Xu J, et al. (2008) Cancer statistics. CA Cancer J Clin 58: 71-96.

25. Notterman DA, Alon U, Sierk AJ, Levine AJ (2001) Transcriptional gene expression profiles of colorectal adenoma, adenocarcinoma, and normal tissue examined by oligonucleotide arrays. Cancer Res 61: 3124-3130.

26. Hou T, Wang J, Zhang W, Xu X (2007) ADME evaluation in drug discovery. 6. Can oral bioavailability in humans be effectively predicted by simple molecular property-based rules?. J Chem Inf Model 47: 460-463.

27. Cruciani G, Pastor M, Guba W (2000) VolSurf: a new tool for the pharmacokinetic optimization of lead compounds. Eur J Pharm Sci 11: S29-S39.

28. Pearlman RS, Smith KM (1999) Metric validation and the receptor-relevant subspace concept. Journal of Chemical Information and Modeling 39: 28-35.

29. Homer N, Merriman B, Nelson SF (2009) BFAST: an alignment tool for large scale genome resequencing. PLoS One 4: e7767.

30. Langmead B, Trapnell C, Pop M, Salzberg SL (2009) Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. Genome Biol 10: R25.

31. Trapnell C, Williams BA, Pertea G, Mortazavi A, Kwan G, et al. (2010) Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation. Nat Biotechnol 28: 511-515.

32. McKenna A, Hanna M, Banks E, Sivachenko A, Cibulskis K, et al. (2010) The Genome Analysis Toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data. Genome Res 20: 1297-1303.